REMARKS

This paper is supplemental to the Amendment and Response Under 37 CFR 1.116 filed with a Request for Continued Examination (RCE) and a Petition For Revival of an Application Abandoned Unintentionally pursuant to 37 C.F.R. §1.137(b) with the U.S. Patent Office on October 3, 2007.

Claims 1-2, 5-8, 11-12, 16-18, 23, and 25 are amended. No claims are canceled or added. As a result, claims 1-19 and 23-25 are now pending in this application.

Amendments to the Title and Specification

The title has been amended to more accurately reflect the claimed subject matter by substituting "Systems, Methods, and Apparatus" for "System and Method".

In the specification, several occurrences of "condition check module 175" have been replaced by "combination function module 160" in order to clarify the written description and make it internally consistent. These replacements are in the paragraphs beginning on page 9, line 18; page 10, line 10; page 11, line 15; and page 12, line 21.

Support for these amendments may be found, for example, in the paragraph beginning on page 6, line 15, which includes the following description:

> As previously discussed, in order for an SIMD capable processor, such as processor 165, to effectively be able to manipulate multiple pieces of data (100 - 115) it is necessary to logically combine the results of mathematical operations shown in arithmetic flags 100, 125,130, and 135. **This is accomplished by the combination function module 160 utilizing the methods and operations illustrated and further discussed in reference to FIGs. 3 to 6.** The results of the combination function performed by the combination function module 160 is a combined arithmetic flag variable 170. **Thereafter, a condition check module 175 is utilized to determine the next operation to perform based upon the combined arithmetic flag variable 170.** [emphasis added]

From the foregoing description, it should be clear that the combination function module 160, and not the condition check module 175, performs the operations described in the paragraphs beginning on page 9, line 18; page 10, line 10; page 11, line 15; and page 12, line 20.

Additional support for this may be found, for example, in original claims 1 and 2. In original claim 1, the combination function module examines the plurality of arithmetic flags, determines the field size, and combines the plurality of flags into a single combined arithmetic flag variable. In original claim 2, the condition check module determines the status of the combined arithmetic flag variable and causes the processor to execute an appropriate operation based on the status.

In addition to the amendments mentioned above, several minor typographical errors have been corrected in the specification.

No new matter has been introduced through these amendments to the specification.

Amendments to the Abstract

The Abstract has been amended to more accurately describe the disclosed subject matter. Also, several grammatical and typographical errors have been corrected. No new matter has been introduced through these amendments to the Abstract.

Amendments to Claims 1-2, 5-8, 11-12, 16-18, 23, and 25

Claims 1-2, 5-8, 11-12, 16-18, 23, and 25 have been amended. No new matter has been introduced.

In claim 1, "to manage" has been substituted for "for combining"; support may be found, for example, on page 1, lines 6-7. The phrase "wherein an M-bit set of arithmetic flags is associated with each of a plurality of data items of varying field sizes within an N-bit word, M and N being positive integers" has been inserted into the preamble; support may be found, for example, on page 5, lines 17-24. The phrase "to examine a word comprising a plurality of sets of arithmetic flags" has been substituted for "that examines a plurality of arithmetic flags"; "to determine a data item" has been substituted for "determines"; "for the word" has been substituted for "of the plurality of arithmetic flags"; "data item" has been inserted; "to" has been substituted for "will"; "of the sets within the word" has been inserted; "of M bits" has been inserted; "result" has been inserted; "the plurality of data items" has been substituted for "a plurality of data items"; and "a processor" has been substituted for "the processor".

In claims 2 ,11, and 16, "result status" has been substituted for "status" (two occurrences).

In claim 5, "wherein" has been inserted.

In claims 6, 7, and 17, "result status" has been substituted for "status".

In claims 8 and 18, "on" has been inserted.

In claim 12, "wherein" has been inserted, and "result status" has been substituted for "status".

In claim 23, similar, if not identical, amendments have been made as were made to claim 1.

In claim 25, the dependency has been switched from claim 23 to claim 24, and "include" has been substituted for "includes".

The amendments to the claims are made to satisfy Applicant's preferences, not necessarily to satisfy any legal requirement(s) of the patent laws, and they are not intended to limit the scope of equivalents to which any claim element may be entitled.

Rejection of Claims 1-4, 6-9, 12-14, and 17-22 Under 35 U.S.C. §102(e) as Anticipated by Wilson

Claims 1-4, 6-9, and 17-19 were rejected under 35 U.S.C. §102(e) as being anticipated by Wilson (U.S. 6, 530, 012) (hereinafter "Wilson"). Applicant assumes from the Examiner's remarks that claims 12-14 were also to be included in this rejection.

Applicant does not admit that Wilson is prior art and reserves the right to swear behind Wilson as provided for under 37 C.F.R. §1.131.

The rule under 35 U.S.C. §102 is well settled that "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2D 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). MPEP §2131.

Applicant asserts that claims 1-4, 6-9, 12-14, and 17-19 are not anticipated by Wilson, because Wilson fails to teach each of the elements included in independent claims 1, 7, 12, 17, and 23, for example.

For example, independent claims 1 and 23, as currently amended, recite:

> a combination function module to examine a word comprising a plurality of sets of arithmetic flags, to determine a data item field size for the word, and based on the determination of the data item field size to **logically combine the plurality of arithmetic flags of the sets within the word into a single combined arithmetic flag variable of M bits**, wherein the plurality of arithmetic flags represent a result status of the plurality of data items after a mathematical operation is performed by a processor on the plurality of data items [emphasis added]

There is no disclosure in Wilson of "logically combine the plurality of arithmetic flags of the sets within the word into a single combined arithmetic flag variable of M bits, wherein the plurality of arithmetic flags represent a result status of the plurality of data items after a mathematical operation is performed by the processor on the plurality of data items", as recited in independent claims 1 and 23.

In contrast, Wilson at column 4, lines 43-45 states:

> The general purpose registers also include a condition code register (CCreg) and a test register TSTreg which are discussed in more detail in the following.

With regard to the condition code register, Wilson at column 7, lines 48-55 states:

> A condition code generator 48 also examines the results of each addition and generates condition codes for the side of the machine where the instruction is being executed to be held in the condition code register CCreg. The condition generator 48 always generates eight condition codes for each side of the machine regardless of the degree of packing of the source operands, as described more fully in the following.

And further, Wilson at column 7, line 60 through column 8, line 1 states:

> The condition code generator 48 generates condition codes CCX0 to CCX7 **by considering the results of the addition operations which were carried out on each packed object in the source registers** and determining from those operations the values of N,Z,C and V which are the bits defining each condition code. In the example of FIG. 6, a different condition code can be generated for each condition code location CCX0 . . . CCX7 in the condition code register. [emphasis added]

Thus, Wilson discloses a condition code register (FIG. 5) and a condition code generator (48, FIG. 7) that generates condition codes by considering the results of addition operations carried out on packed objects in the source registers (SRC1 and SRC2, FIG. 7). However, there is no teaching in Wilson of logically combining any of these generated results with any other of the results. Thus, Wilson fails to teach **"logically combine the plurality of arithmetic flags of the sets within the word into a single combined arithmetic flag variable of M bits**, wherein the plurality of arithmetic flags represent a result status of the plurality of data items after a mathematical operation is performed by a processor on the plurality of data items" [emphasis added], as recited in independent claims 1 and 23. Thus, Wilson fails to teach each of the elements recited in claims 1 and 23, and therefore claims 1 and 23 are not anticipated by Wilson.

In further examples of elements recited in the claims and not taught by Wilson, independent claims 7 and 12 each recite:

> **logically** combining the plurality of arithmetic flags based on a function selected when a combination process is selected [emphasis added]

For reasons analogous to those discussed above with respect to independent claims 1 and 23, Wilson fails to teach "logically combining the plurality of arithmetic flags based on a function selected when a combination process is selected", as recited in independent claims 7 and 12. Thus, Wilson fails to teach each of the elements recited in claims 7 and 12, and therefore claims 7 and 12 are not anticipated by Wilson.

In a further example of elements recited in the claims and not taught by Wilson, independent claim 17 recites,

> extracting the plurality of arithmetic flags based on the field size, the plurality of arithmetic flag associated with a single selected data item; and
>
> storing a result of the extracting of the plurality of arithmetic flags in a destination register for access by the processor.

Applicant is unable to find in Wilson a teaching of these elements as recited in claim 17 and as quoted above. Thus, Wilson fails to teach each of the elements recited in claim 17, and therefore claim 17 is not anticipated by Wilson.

Claims 2-4 and 6 depend from claim 1, and so include all of the elements recited in claim 1. Claims 8-9 depend from claim 7, and so include all of the elements recited in claim 7. Claims 13-14 depend from claim 12, and so include all of the elements recited in claim 12. Claims 18-19 depend from claim 17, and so include all of the elements recited in claim 17. Claims 24-25 depend from claim 23, and so include all of the elements recited in claim 23. For at least the reasons stated above, with respect to independent claims 1, 7, 12, 17, and 23, and the additional elements included in claims 2-4, 6, 8-9, 13-14, and 18-19, these claims are also not anticipated by Wilson.

In an example of additional elements recited in dependent claims and not taught by Wilson, claims 6, 11, and 16 include a listing of the result status of a plurality of operations that may be performed on the combined arithmetic flag variable by the condition check module. Wilson fails to disclose these operations, as recited in claims 6, 11, and 16.

In an attempt to supply these elements missing from Wilson, the Final Office Action on pages 7-10 refers to Table 1 in column 6 of Wilson. However Wilson at column 5, line 62 through column 6, line 7 states:

> The 8-bit field of each Treg is as shown in FIG. 4.
> The Condition field (bits 0 to 3) applies to all predicated instructions. It holds a 4 bit test code to allow for conditions to be tested. As discussed in more detail later, for instructions on packed objects, the condition applies to all the lanes on a per lane basis.
> The four condition flags are: N (Negative flag -- bit 3) Z (Zero flag -- bit 2) C (Carry flag -- bit 1) V (Overflow flag -- bit 0)
> **These four bits give rise to 16 test conditions** (see Table 1).
> [emphasis added]

Thus, Table 1 of Wilson discloses test conditions associated with FIG. 4. FIG. 4 represents a test register Treg, and therefore, any "test conditions" described in Table 1 of Wilson relate to test register Treg. Wilson describes how these test conditions are set at column 5, lines 50-53, which states:

> The Treg bytes are programmable and are set to the required values prior to instruction execution. The test register is used to allow conditional execution of instructions.

Thus, these bytes in test register Treg, referred to in Table 1 of Wilson, are programmable and set to the required values prior to instruction execution. However, there is no disclosure in Wilson that teaches a result status associated with a plurality of data items after a mathematic operation is performed by the processor on the plurality of data item, as recited in claims 6, 11, and 16. Therefore, Wilson fails to disclose these additional elements and so fails to anticipate claims 6, 11, and 16.

Applicant respectfully requests withdrawal of the 35 U.S.C. §102 (e) rejection of claims 1-4, 6-9, 12-14, 17-19.

Rejection of Dependent Claims 5, 10-11, and 15-16 under 35 U.S.C. §103(a) as Unpatentable over Wilson in view of Bindloss

Dependent claims 5, 10-11, and 15-16 were rejected under 35 U.S.C. §103(a) as being unpatentable over Wilson in view of Bindloss et al. (U.S. 5, 778, 241).

To establish a *prima facie* case of obviousness under 35 U.S.C. §103, the prior art reference (or references when combined) must teach or suggest every limitation of the claim. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA, 1974). MPEP §2143.

Claim 5, which is dependent from claim 1, includes all of the elements of claim 1, and is therefore patentable over Wilson for at least the same reasons as stated above for claim 1. Bindloss does not supply the elements of claim 1 that are missing from Wilson. Thus, the proposed combination of Wilson and Bindloss fails to teach or suggest each of the elements included in claim 5.

Claim 10-11, which are dependent from claim 7, include all of the elements of claim 7, and are therefore patentable over Wilson for at least the same reasons as stated above for claim 7. Bindloss does not supply the elements of claim 7 that are missing from Wilson. Thus, the proposed combination of Wilson and Bindloss fails to teach or suggest each of the elements included in claims 10-11.

Claim 15- 16, which are dependent from claim 12, include all of the elements of claim 12, and are therefore patentable over Wilson for at least the same reasons as stated above for claim 12. Bindloss does not supply the elements of claim 12 that are missing from Wilson.

Thus, the proposed combination of Wilson and Bindloss fails to teach or suggest each of the elements included in claims 15-16.

For at least the reasons stated above, Applicant respectfully requests withdrawal of the 35 U.S.C. §103 rejection of claims 5, 10-11, and 15-16.

CONCLUSION

Applicant respectfully submits that claims 1-19 and 23-25 are in condition for allowance, and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's attorney Ann M. McCrackin (located in Minneapolis, Minnesota) at (612) 349-9592 or Applicant's below-signed attorney (located in Phoenix, Arizona) if there are any questions regarding this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 19-0743.

Respectfully submitted,

NIGEL C. PAVER

By his Representatives,
SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. Box 2938
Minneapolis, Minnesota 55402
(602) 298-8920

By / Walter W. Nielsen /
Walter W. Nielsen
Reg. No. 25,539